A First Characterization of Anycast Traffic from Passive Traces

A First Characterization of Anycast Traffic from Passive Traces

Danilo Giordano[1], Danilo Cicalese[3], Alessandro Finamore[2],
Marco Mellia[1], Maurizio Munafò[1], Diana Zeaiter Joumblatt[3], Dario Rossi[3]
[1]Politecnico di Torino - first.last@polito.it
[2]Telefónica Research - first.last@telefonica.com
[3]Telecom ParisTech - first.last@enst.fr

***Abstract*—IP anycast routes packets to the topologically nearest server according to BGP proximity. In the last years, new players have started adopting this technology to serve web content via Anycast-enabled CDNs (A-CDN). To the best of our knowledge, in the literature, there are studies that focus on a specific A-CDN deployment, but little is known about the users and the services that A-CDNs are serving in the Internet at large.**

This prompted us to perform a passive characterization study, bringing out the principal A-CDN actors in our monitored setup, the services they offer, their penetration, etc. Results show a very heterogeneous picture, with A-CDN empowered services that are very popular (e.g., Twitter or Bing), serve a lot of different contents (e.g., Wordpress or adult content), and even include audio/video streaming (e.g., Soundcloud, or Vine). Our measurements show that the A-CDN technology is quite mature and popular, with more than 50% of web users that access content served by a A-CDN during peak time.

I. Introduction

IP anycast allows a group of geographically distributed servers to share a common IP address. When a client contacts an IP anycast server, the packets are thus routed at the network layer to the closest address according to the BGP routing distance.

Deploying an IP anycast service is relatively easy and introduces several advantages such as load balancing between the servers, DDoS mitigation, and increases the reliability. This is the primary reason of its adoption in multiple *stateless* services running on the top of UDP, e.g., DNS root and top level domain servers, 6-to-4 relay routers, multicast rendezvous points, and sinkholes.

When considering *stateful* services, the usage of IP anycast has been discouraged primarily due to its lack of control and awareness for the server and network load. Indeed, IP anycast relies on BGP routing, meaning that any routing change could re-route the traffic to another server. This could break any stateful service, e.g., causing the abortion of TCP connections, and could cause the dropping of any application layer state. Moreover, the relatively slow convergence of routes and the purely destination based routing in IP make difficult design reactive systems where traffic can be arbitrarily split among multiple surrogate nodes. This initially led the Content Delivery Network (CDN) companies to use other load-balancing techniques, i.e., leveraging DNS or HTTP to direct the customer requests to the best surrogate server [15]. Over the years however, several studies proposed techniques to overcome these issues, showing that it is possible to leverage anycast for connection oriented services [1], [2], [14]. This let companies to deploy Anycast-enabled CDNs (A-CDN), in which multiple surrogate servers use the same IP address whose reachability is managed by IP anycast. CacheFly [1] was among the A-CDN pioneers, followed then by other companies including Edgecast and CloudFlare to name the most popular ones. Our own recent work [8] shows that A-CDN technology is mature and readily available, with Internet service providers (e.g. AT&T Services), social networks (e.g. Twitter) and cloud providers (e.g Microsoft) having adopted IP anycast to provide stateful services.

However, to the best of our knowledge, previous works that focused on A-CDNs limitedly leveraged *active measurements* to discover geolocation of anycast replicas, or to benchmark the performance of some specific deployment (see Sec.II for more details). These works show considerable interest in the topic, but they fall short in providing an actual characterization of A-CDNs from the end-user point of view: in other words, how much traffic do they serve? which services do they offer? etc. In order to answer these questions, we leverage *passive measurements*, and offer a characterization of traffic served by A-CDNs in real networks. We use the IP anycast prefixes discovered by the largest census in [8], where about 1600 /24 subnets have been discovered hosting IP anycast servers. This list is compiled using active probing. Given this list, with the aim at providing a first characterization of modern usage of A-CDNs, we use traffic traces from approximately 20,000 households collected from a large European ISP for the entire month of March 2015 (when the census was performed). This large dataset allows us to obtain a snapshot of (i) how popular A-CDNs are, (ii) which services they support, (iii) which are the characteristics of traffic they serve and (iv) their proximity and affinity performance.

We summarize our main findings as follows:

- We confirm IP anycast to be not anymore relegated to the support of connectionless services: in a month we observe over 16,000 active anycast servers contacted via TCP, mapping to more than 92,000 hostnames.
- While hard to gauge via passive measurement, and despite the limited scope that our single vantage point offers, we in general observe stable paths, with only a handful changes during one month.

[1] http://www.cachefly.com/about.html

- Both large players like Edgecast or Cloudflare, and smaller but specialized A-CDNs are present: content served include heterogeneous services such as Twitter Vine, Wordpress blogs, TLS certificate validation and BitTorrent trackers. Footprint of A-CDN is also very different, with some being pervasive enough to have servers at few ms from customers, while others have fewer replica nodes that turn out to be more than 100ms far away.
- In our datasets, A-CDNs are fairly popular: 50% of users encounter at least an A-CDN server during normal web activity. Thus, penetration of A-CDN is already very relevant.
- Most of TCP connections last few tens of seconds and carry a relatively small amount of bytes; surprisingly however, we see video and audio streaming services being supported by A-CDNs, whose TCP flow last for several hours. The latter could be affected by sudden routing changes that could break TCP connections. However, given the infrequent occurrence of such events, it is hard to measure (and suffer from) it in practice.

We hope the facts and figures highlighted in this paper contribute to the knowledge of operational IP anycast deployments, and of A-CDN in particular. To further assist the research community in the understanding of modern IP anycast, we make the dataset used in this paper available to the interested reader.[2]

II. Related work

We can categorize the large body of work that investigates IP anycast in three coarse categories: (i) anycast improvement proposals, (ii) studies of anycast performance of specific deployments, (iii) broad assessment of anycast adoption. As for (i), several studies, starting from seminal work such as [17] and culminated recently in [14], propose architectural improvements to address the performance shortcomings of IP anycast in terms of scalability and server selection. They differ from our work since we instead aim at studying the broad variety of existing deployments in the wild.

Concerning (ii), several works focus on deeply studying specific aspects of anycast performance. With few exceptions [7], and in spite of evidence [18] that anycast is successfully used beyond the DNS realm, most of the work targets DNS as the anycast service of interest. The typical key performance indicators include server proximity [3], [7], [11], [19], [21], client-server affinity [3], [4], [6], [7], [19], [21], server availability [3], [16], [21], and load-balancing [3], [4]. All the methodologies involve active measurements, and thus they are orthogonal to this work. Closest work to ours under this perspective are [4], [19], which base their investigations on passive measurement methodologies. In both cases, authors' collection point is located at the anycast servers, so that they obtain a complete view of all user requests served by specific single server. Conversely, our vantage point is located close to the customers in an ISP network, and it allows us to gather a complementary view of *all anycast services*, albeit from a possibly limited set of users and a single country.

Finally, concerning (iii) there has been a renewed interest in a broad assessment of anycast services, with techniques capable of detecting anycast usage [20], or even enumerate [12] and geolocate [8], [10] anycast replicas. All these techniques perform active measurement to infer some properties of anycast services: in particular, [8], [20] perform censuses of the IPv4 address space to find evidence of anycast deployments via active measurement on the data plane. Despite these studies provide a broad characterization of anycast deployment, they however fail in capturing how popular such services are, how much traffic they attract, and which applications they serve. These are precisely the questions we address in this paper, taking the census of IP anycast subnet discovered in [8] as a starting point, and using a passive methodology to complement and refine the general picture that can be gathered with active measurements.

III. Methodology

A. Anycast subnet lists

In a nutshell, our workflow first extract the subset of flows that are directed to anycast servers, and then characterize the traffic they exchange with actual internet users by leveraging passive measurements. To identify anycast servers, we rely on the *exhaustive* list of /24 subnet prefixes that result to host at least one anycast server according to the IP censuses we performed in March 2015 [8].[3] As described in [8], we compile an *exhaustive* list of 1696 anycast subnets, by simultaneously pinging an IP/32 from all valid IP/24 networks from PlanetLab probes. The scan runs on all IP address range. Next, we ran the anycast detection technique developed in [10] to identify IP/32 anycast addresses (i.e., located in more than one geographical location). Intuitively, we look for RTT measurements that violates the propagation time constraint from different vantage points. For instance, when pinging an host from two probes, the sum of the RTT measurements to the same server IP must be higher than the propagation time from the two probes. We flag as anycast network any network having at least two locations in our censuses. From this list, we extract a more *conservative* set of 897 subnets, having at least five distinct locations. Since the conservative set is biased toward larger and likely more popular deployments, we expect the conservative set to yield an incomplete but comprehensive picture of IP anycast. In the following, we use this list to inform the passive monitor about the subnets of interest.

B. Passive monitor

We instrumented a passive probe at one PoP of an operational network in an European country-wide ISP.[4] The probe runs Tstat [13], a passive monitoring tool that observes packets flowing on the links connecting the PoP to the ISP backbone network. The probe uses professional Endace cards

[2]Researchers interested in data used in this paper are invited to contact us.

[3]Recall that BGP announced prefixes have a minimum granularity of a /24 subnet. Thus, an anycast address range cannot be smaller than a /24 range.

[4]Results from other vantage points in other PoPs are practically identical. For easy of presentation, we focus on one PoP in this paper.

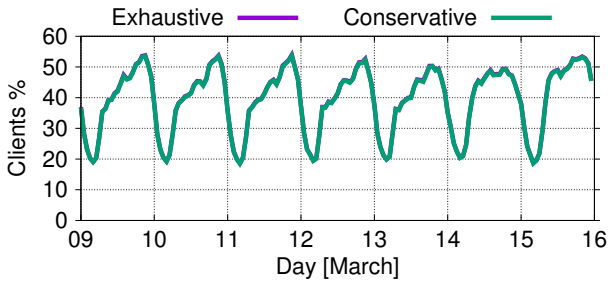


Fig. 1: Percentage of clients that contact at least one A-CDN server in each 1h time slot, both curves overlap.

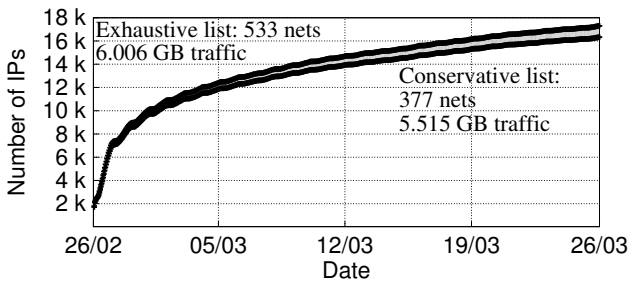


Fig. 2: Cumulative number of distinct servers encountered over the month.

to guarantee all packets are efficiently exposed in user-space for processing. No sampling is introduced, and the probe is able to process all packets [22]. Tstat rebuilds in real time each TCP and UDP flow in real time, tracks it, and, when the flow is torn down or after an idle timer, it logs more than 100 statistics in a simple text file. For instance, Tstat logs the client and server IP addresses[5], the application (L7) protocol type, the amount of bytes and packets sent and received, the TCP Round Trip Time (RTT), etc.

Tstat implements DN-Hunter [5], a plugin that annotates each TCP flow with the server Fully Qualified Domain Name (FQDN) the client resolved via previous DNS queries. For instance, assume a client would like to access to *www.acme.com*. It first resolves the hostname into IP address(es) via DNS, getting 123.1.2.3. DN-Hunter caches this information. Then, when later the same client opens a TCP connections to 123.1.2.3, DN-Hunter returns *www.acme.com* from its cache and associate it to the flow. Our vantage points observe all traffic generated by clients, including DNS traffic directed to local resolvers. Client DNS cache is rebuild in Tstat, resulting in more than 95% accuracy [5]. This is particularly useful for unveiling *services* accessed from simple TCP logs. This is useful since it unveils the service being offered by the server having IP address 123.1.2.3, even in presence of encrypted (e.g., HTTPS) or proprietary protocols. [6]

For this study we leverage a dataset collected during the whole month of March 2015. It consists of 2 billions of TCP flows being monitored, for a total of 270 TB of network traffic. 1.4 billion connections are due to web (HTTP or HTTPS) generating 209 TB of data. More important, we observe more than 20,000 ISP customers active over the month, which we identify via the static anonymized client IP address[7]. All traffic generated by any device that accesses the internet via the home gateway is thus labeled by the same client IP address. This includes PCs, smartphone, Tablets, connected TV, etc. that are connected via WiFi or Ethernet LAN to the home gateway.

Among the many measurements provided by Tstat for each TCP flow, we focus only on: (i) The minimum Round-Trip-Time (RTT) between the Tstat probe and the server; (ii) The amount of downloaded bytes; (iii) The application layer protocol (e.g., HTTP, HTTPS, etc.); and (iv) The FQDN of the server the client is contacting. These metrics are straightforward to monitor, and details can be found in [5], [13].

IV. Anycast at a glance

A. Temporal properties

We first provide an overall characterization of the anycast traffic. Not surprisingly, we observe that all anycast UDP traffic is labeled as DNS protocol – which we avoid investigating given the literature on anycast DNS. More interestingly, we observe a sizeable amount of anycast traffic carried over TCP: overall, almost 59 million TCP connections are managed by anycast servers. Those correspond to approximately 3% of all web connections and 4% of the aggregate HTTP and HTTPS volume, for a total of 6 TB of data in the entire month. Definitively a not-negligible amount of traffic, especially when considering the relatively small number of /24 anycast subnets.

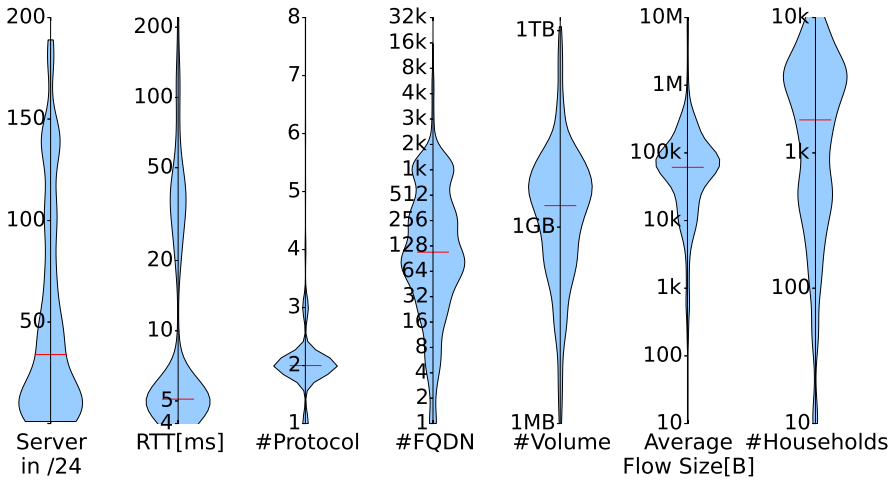
The large majority of traffic is directed to TCP port 80 or 443, that the DPI classifier labels as HTTP and SSL/TLS, respectively. This suggests that hosted services are indeed offered by A-CDNs and served over HTTP/HTTPS. A minority of the traffic (less than 1% of all anycast traffic) is instead related to some protocols for multimedia streaming, email protocols, Peer-to-Peer traffic, or DNS over TCP. We will provide further details when digging into some selected examples.

Results confirm the footprint of anycast traffic, and A-CDN in particular. To corroborate this, Fig. 1 shows evolution during one-week of the percentage of active customers that have encountered at least one anycast server during their normal web browsing activities (the ratio is computed at hourly intervals, normalizing over the number of client active in that hour). Besides exhibiting the day/night pattern due to the different nature of services running on the network with fewer services served over anycast at night, the figure shows that at peak time the probability to contact at least one anycast server is higher than 50%. Notice that Fig. 1 reports the probabilities according to both the exhaustive and the

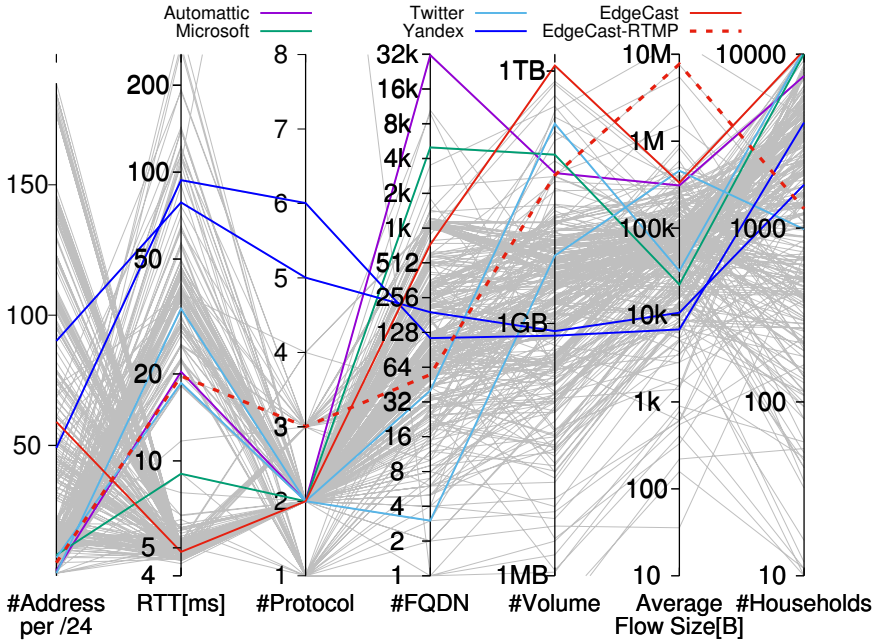
[5]We take care of obfuscating any privacy sensitive information in the logs. Customer IP addresses are anonymised using irreversible hashing functions, and only aggregate information is considered. The deployment and the information collected for this have been approved by the ISP security and ethic boards.

[6]Collisions may be present, e.g., when the same client contacts *mail.acme.com* which is hosted by the same server 123.1.2.3. Since in this work we are interested on which services a given server hosts, collisions are not critical, e.g., we can discover that 123.1.2.3 serves both *www.acme.com* and *mail.acme.com*.

[7]The ISP adopts a static addresses allocation policy, so that each customer home gateway is uniquely assigned the same static IP address.



(a) violin plots



(b) Parallel coordinate plot. For the ease of readability, single curve plots can be found at [9]

Fig. 3: Anycast at a glance[8]

conservative lists: these curves cannot be distinguished as they perfectly overlap, which hints to the fact that the conservative list is, as expected, comprehensive enough for our purposes. This clearly may change on vantage points located in different countries. However, for the purpose of this work, we prefer to take a conservative approach.

Fig. 2 reports the cumulative number of unique IP anycast addresses observed over time, again for both the conservative and exhaustive lists. In total, over 16,000 distinct IP addresses are observed during the whole month for the conservative list. The picture is additionally annotated with the traffic volume exchanged with such servers: notice that despite the exhaustive list is twice as big as the conservative one, the number of servers contacted and bytes exchanged are fairly similar. This happens since the major A-CDN players are present in the conservative list, to which we thus limitedly focus on the following. Notice also that the number of distinct servers encountered over the month quickly grows during the first days, during which most popular services and servers are contacted.

B. *Service diversity*

We now provide an overall picture of A-CDN diversity with a dual violin plots (top) and parallel coordinate (bottom) representation in Fig.3. Violin plots compactly represent the marginals for some metrics of interest, whereas parallel coordinate plots allow us to grasp the correlation between these metrics for some specific deployments. On both plots,

TABLE I: Dataset summary

/24 subnet	Owner	IP/32	Vol. [GB]	Flows [k]	Users	FQDN	Protocols	Content/Service
93.184.220.0	EdgeCast-1	105	357	8,018	10,626	3,611	HTTP/s	generic
199.96.57.0	**Twitter-generic**	7	219	7,318	10,508	40	HTTP/s	twitter, vine
68.232.34.0	**EdgeCast-2**	59	1,071	3,484	10,490	736	HTTP/s	microsoft, spotify
68.232.35.0	EdgeCast-3	104	480	5,059	10,354	904	HTTP/s	twitter, gravatar, tumblr
94.31.29.0	NetDNA	73	80	1,292	10,218	609	HTTP/s	generic
93.184.221.0	EdgeCast-4	49	708	2,031	10,155	1,467	HTTP/s	generic
204.79.197.0	**Microsoft**	8	93	4,508	10,044	5,088	HTTP/s	bing, live, microsoft
205.185.216.0	Highwinds-1	2	180	1,411	9,705	267	HTTP/s	generic
108.162.232.0	CloudFlare-1	13	53	550	9,274	14	HTTP	ocsp certificates
178.255.83.0	Comodo	5	3	601	8,837	65	HTTP	ocsp certificates
192.0.72.0	**Automattic**	2	57	199	7,477	32,037	HTTP/s	wordpress
108.162.206.0	CloudFlare-2	122	14	246	4,695	465	HTTP/s, Torrent	P2P trackers, generic
213.180.193.0	**Yandex-2**	49	0.7	105	4,031	114	HTTP/s SMTP	yandex
213.180.204.0	**Yandex-1**	90	0.7	76	1,771	191	HTTP/s, SMTP	yandex
93.184.222.0	**EdgeCast-RTMP**	5	53	8	1,289	55	RTMP, HTTP	soundcloud, video
199.96.60.0	**Twitter-vine**	1	6	14	983	3	HTTP/s	vine
Total	Exhaustive	17,298	6,006	58,885	10,830	120,151	-	-
Total	Conservative	16,329	5,515	54,045	10,828	117,768	-	-

we select the following axes: (i) the number of active servers in the /24 subnet, (ii) the average minimum RTT for any server in that /24, (iii) the number of distinct protocols, (iv) the number of distinct FQDNs/services, (v) the total amount of bytes served during the whole period, (vi) the average flow size in bytes, and (vii) the number of households that contacted one server in the /24.

In more details, violin plots of Fig.3 are an intuitive representation of the Probability Density Function (PDF): the larger their waist is, the higher is the probability of observing that value; red bars are a further visual reference, corresponding to the median of the distribution. Overall, the plot shows that most of /24 host few servers, which are in general quite close to the PoP (RTT<10 ms), use 2 or at most 3 protocols (HTTP or HTTPS mostly). Diversity starts to appear in the number of served FQDNs – with some /24 being used for a handful services, while others serve several thousands. Served volume varies widely. Similarly, while only half of flows exceed 50 kB, and most are below 1 MB, flow size peaks up to several hundreds MB (see Sec.III for more details). Considering popularity, some /24 are used by several thousands end-users, others by less than 10.

Parallel coordinates instead allow the observation of a specific deployment: each line represent a /24 subnet, and the "path" among the vertical axes highlights the characteristic of that A-CDN over different dimensions. We report most[9] /24 with light gray color, and additionally we highlight some of them using different colors. First, observe that the wide dispersion of the light gray paths testifies great diversity across A-CDN deployments.

Next, observe per-deployment dispersion of the few selected /24. For the sake of illustration, consider the Automattic curve, which is the A-CDN that serves websites hosted by Wordpress: it can be seen that the two active servers found in the /24 are located at about 20 ms from the PoP. They use both HTTP and HTTPS, for a total of more than 32,000 FQDNs. Total volume accounts for 20 GB during the month, transferred over flows that are 200 kB long on average. At last 7400 users (37%) accesses some content hosted by Automattic. Without going into details for lack of space, we have highlighted telling examples such as: Twitter A-CDN (which serve few domains); Microsoft A-CDN (bing.com and live.com services, for a total of more than 5000 FQDNs); one /24 of EdgeCast as an example of a generic A-CDN; a specialized EdgeCast platform serving audio and video streams over Real Time Media Protocol (RTMP), see the red dashed line. Finally, we selected two /24 belonging to Yandex, the most popular search engine and social platform in Russia, that are however not among the most popular in the geographic region of our vantage point. It clearly appears that these examples, which we more deeply investigate in the rest of the paper, are representative of quite diverse anycast deployments. This makes it difficult to highlight common trends, e.g., we observe popular A-CDN whose RTT is quite large, and unpopular A-CDN whose RTT is instead much smaller.

[9]To reduce visual cluttering, we report in light gray color the subset of /24 that served at least 1000 flows and 10 distinct households during a month.

V. Selected anycast deployments

A. Candidate selection

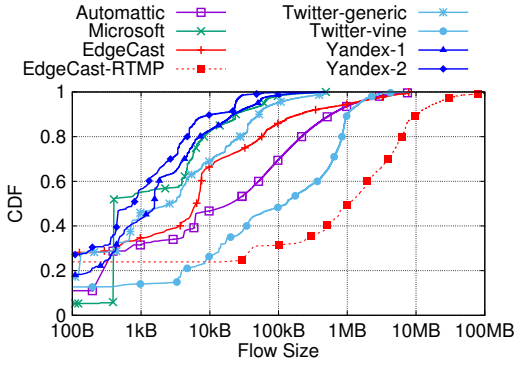
Tab. I offers details for some selected deployments. Rows highlighted in bold font refers to the same subnets previously highlighted in Fig. 3.

For each /24 subnet, Tab. I lists the Owner (i.e., the organization managing it as returned by Whois), the number of distinct server addresses that have been contacted at least once, the total volume of bytes served, the number of flows, of users, and of distinct FQDNs. The last two columns report the most prominent protocols and services the A-CDN offers.

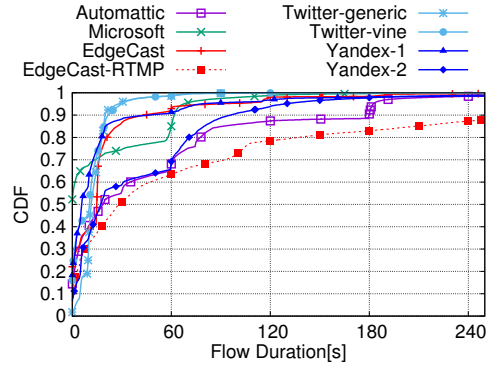
The table, which also serves as a summary of our anycast dataset, comprises the top-10 most popular /24 A-CDN prefixes and subnets (top part). To avoid an excessive bias toward only popular services (where HTTP and HTTPS are predominant), we additionally report some manually selected A-CDNs (bottom part).

As it can be observed, the portfolio of services supported by A-CDNs includes email via SMTP, video/audio streaming via RTMP, certificate validation via Online Certificate Status Protocol (OCSP), and even BitTorrent Trackers.

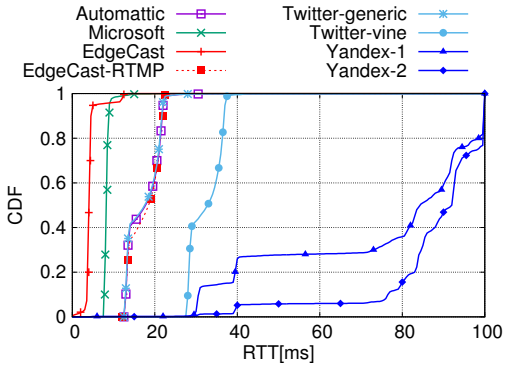
The table precisely quantifies the very heterogeneous scenario early depicted by the Fig. 3. EdgeCast is the major player in our dataset, managing 4 of the top-10 subnets: each of these



(a) Flow size CDF



(b) Flow duration CDF



(c) Round-trip time CDF

Fig. 4: Metrics characterization

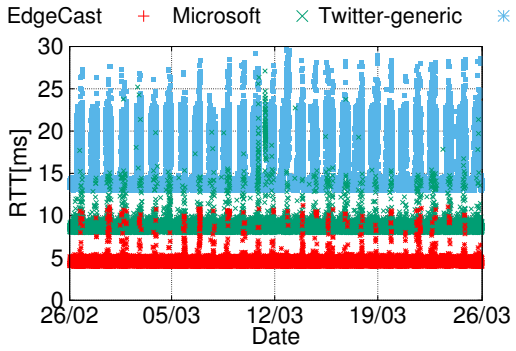


Fig. 5: Stable Situation. Single curve plots can be found at [9]

serves between 350 GB/month and 1 TB/month to more than 10,000 (50%) households in our PoP. This is not surprising, given EdgeCast claims to serve over 4% of the global Internet traffic[10].

Popular A-CDNs includes Microsoft, which directly manages its own A-CDN. It serves Bing, Live, MSN, and other Microsoft.com services. Since it handles quite a small amount of data and flows, we checked if there are other servers not belonging to the Microsoft A-CDN that handle those popular Microsoft service. We found that all of *bing.com* pages and web searches are actually served by the Microsoft A-CDN, while static content such as pictures or map tiles are instead by the Akamai CDN. Thus, Microsoft is using an hybrid solution based on a traditional CDN and its own A-CDN at the same time.

Finally, popular A-CDN services include Highwinds and Comodo. Highwinds offers video streaming for advertisement companies, and images for popular adult content websites (notice the relative longer lived content). Instead, Comodo focuses its business on serving certificate validations using OCSP, Online Certificate Status Protocol, services (with lot of customers who fetch little information).

Overall, major A-CDN players serve thousands of FQDNs including very popular web services like Wordpress, Twitter, Gravatar, Tumblr, Tripadvisor, Spotify, etc. This explains why about 1 out of 2 end-users likely contacts at least one A-CDN server during her navigation. Most FQDNs are uniquely resolved to one IP address – but the same IP address serves multiple FQDNs. Interestingly, this behaviour is shared among most of the studied A-CDNs, meaning that they purely rely on anycast routing for load-balancing. An exception is CloudFlare's A-CDN which uses also DNS load-balancing. Cloudflare uses up to 8 IP addresses in the same /24 to serve the same FQDN. For lack of space, we are not able to report here the results of a complementary active measurement study that we report at [9].

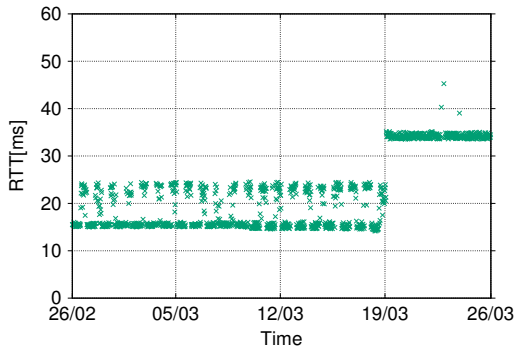
B. Per-deployment view

For the selected deployments, we details the cumulative distribution function (CDF) of some interesting metrics. The CDF is computed considering all the flows being served by the same /24 subnet.

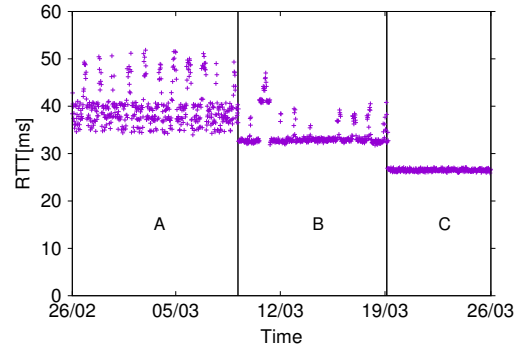
As for the metrics of interest, we report the CDF of flow size (Fig. 4(a)), duration (Fig. 4(b)), and Round Trip Time (Fig. 4(c)). Fig. 4(a) shows how the size of the content hosted by A-CDNs varies across deployments (the different supports only partly overlap), and also between flows of the same deployment (the support is large and, with few exceptions, there is no typical object size). In general served objects are shorter than 1 MB, with the notable exception of audio and video streams served by EdgeCast specialized deployment that support RTMP streaming. In this case, flows are larger than 100 MB.

The small amount of data is reflected on the TCP flow duration CDF. Indeed, Fig.4(b) shows that flow lifetime is in

[10]http://www.edgecast.com



(a) Event 1: sudden change



(b) Event 2: multiple changes

Fig. 6: RTT changes

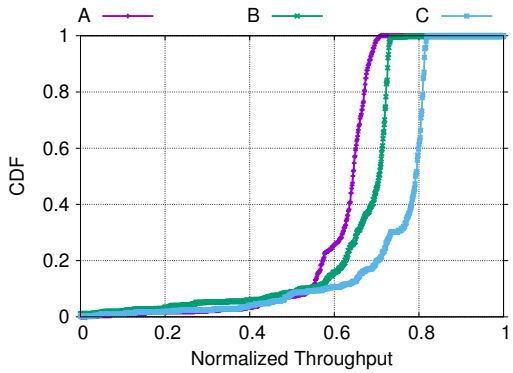


Fig. 7: Event 2: Throughput Implications

general shorter than 180 s, with specific values that reflects typical HTTP server timeouts (multiple of 60 s). Once again, the only exception is the EdgeCast-RTMP deployment, for which over 10% of the TCP flows exceed 5 minutes (visible in the picture), and ranges up to hours (not visible in the picture). Finally, minimum Round Trip Time CDF in Fig.4(c) reveals that the popular A-CDNs have a good footprint (at least in Europe). The only exception is Yandex that have anycast replicas in the eastern Europe and in Russia (which is not surprising due to the language specific content it serves).

Notice how sharp the CDF are for EdgeCast or Microsoft deployment. This suggests that the path to their servers is very short, but also very stable. RTT of other A-CDNs is instead very similar, e.g., EdgeCast-RTMP, Twitter-generic, and Automattic. This suggests that their servers are located in the same place, and reached by the same path. Interestingly, the minimum RTT shows a deviation which suggests the presence of some extra delay for 60% of samples, possibly accounting for some queuing delay due a possibly congested link on that path (which belongs to the Twitter-vine path as well).

C. RTT variation over time

In this section we look for evidences may suggest possible path properties changes. We based our analysis studying the TCP minimum RTT. Intuitively, a *sudden* change in the minimum RTT could highlight a possible sudden change in the routing or network infrastructure. We argue that a dramatic change in the TCP minimum RTT, is likely due to a routing change. Conversely, a *smooth* shift could suggest the presence of queuing delay due to possible congestion on some path links. Passive measurements could only suggest to investigate more deeply of eventual changes, e.g., triggering active measurements to provide a more reliable ground truth to distinguish between hardware improvement and routing changes. For instance, checking HTTP headers could be used to reliably reveal the anycast replicas [8].

Fig. 5 shows the minimum RTT values for each TCP flows over time. Three most used A-CDNs during the entire month of March 2015 are considered. The figure suggests that no sudden changes in the path are visible. Yet, observe the periodic and smooth increase of RTT during the peak time. This could be explained as a congestion events, that are reflected in the smooth changes in the minimum RTT CDF as shown in Fig. 4(c). For these three A-CDN, data suggest stable but possibly congested paths to the anycast addresses.

We investigated the RTT evolution over time of other /24 subnets. We found few cases that we believe could highlights possible sudden changes in the routing plane, possibly affecting server affinity. Fig. 6(a) and Fig. 6(b) report two examples. Plot on the top shows an example of sudden changes affecting another /24 subnet. In this case, the minimum RTT suddenly increases by almost 15ms. Notice also that the path to the longer location is not affected by periodical increases during peak time, suggesting no congestion is present in this second path. Look now at the plot on the bottom. It suggests changes on March 9th, and 19th (with possibly a short change on the 11th). Indeed, minimum RTT properties differ quite significantly and with a sharp change. To study the implication of this from a client perspective, we report the CDF of the throughput for the three distinct periods in Fig. 7. We normalize the throughput between 0 and 1 for ISP privacy motivation. The three distribution show that when the RTT decreases, i.e., the server serving the flow is closed to the users, the throughput improves. Thus, A-CDN changes have an impact on performance as well. We have observed other changes in different /24 networks, not reported here due to lack of space.

We also tried to investigate if changes have implications on TCP connections. In particular, one would expect that an ongoing TCP connections to be abruptly terminated if the routing

change implies a server change as well. We tried to investigate this by correlating number of TCP flows abruptly terminated by a server RST message with possible routing change events. We are not able to observe any clear evidence. Indeed, on the one hand, TCP flows are very short – cfr. Fig. 4(b) – and, on the other hand, changes are sudden and very few. Thus only an handful TCP connections could possibly be involved during a change event. This support the intuition that anycast is indeed well suited for connection oriented services.

In summary, while we observe sharp changes in the anycast path to reach the A-CDN caches, those events are few and occasional, with each different routing configuration that lasts for days. Clearly, deeper investigation is needed to better understand eventual routing changes over the time. In this direction we are trying to exploit other metrics as the Time To Live (TTL) and the Time To First Byte (TTFB) to highlight routing changes. Combining this methodology with active measurements, it could provide a better understanding of routing stability that may affect A-CDN deployments.

VI. Conclusions and Discussion

We presented in this paper a first characterisation of Anycast-enabled CDN. Starting from a census of anycast subnets, we analysed passive measurements collected from an actual network to observe the usage and the stability of the service offered by A-CDNs. Our finding unveil that A-CDNs are a reality, with several players adopting anycast for load balancing, and with users that access service they offer on a daily basis. Interestingly, passive measurements reveal anycast to be very stable, with stable paths and cache affinity properties. In summary, anycast is increasingly used, A-CDNs are prosperous and technically viable.

This work is far from yielding a complete picture, and it rather raises a number of interesting questions such as:

Horizontal comparison with IP unicast. Albeit very challenging, more efforts should be dedicated to compare Unicast vs Anycast CDNs for modern web services. To the very least, a statistical characterization and comparison of the pervasiveness of the deployments (e.g., in term of RTT) and its impact on objective measures (e.g., time to the first byte, average throughput, etc.) could be attempted.

Vertical investigation of CDN strategies. From our initial investigation, we noticed radically different strategies, with e.g., hybrid DNS resolution of few anycast IP addresses, use of many DNS names mapping to few anycast IPs, use of few names mapping to more than one anycast IPs, etc. Gathering a more thorough understanding of load balancing in these new settings is a stimulant intellectual exercise which is not uncommon in our community.

Further active/passive measurement integration. As anycast replicas are subject to BGP convergence, a long-standing myth is that it would forbid use of anycast for connection-oriented services relying on TCP. Given our results, this myth seems no longer holding. Yet, while we did not notice in our time frame significant changes in terms of IP-level path length, more valuable information would be needed from heterogeneous sources, and by combining active and passive measurements.

Acknowledgements

The research leading to these results has received funding from a Google Faculty Research Award and the European Union under the FP7 Grant Agreements no. 318627 (Integrated Project “mPlane”). The work has been partially carried out at LINCS - Laboratory of Information, Networking, and Computer Science (www.lincs.fr). We also thank our shepherd, Benoit Donnet, and the anonymous TMA reviewers for their valuable feedback.

References

[1] Z. Al-Qudah, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van der Merwe. Anycast-aware Transport for Content Delivery Networks. In *Proc. ACM WWW*, 2009.

[2] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van Der Merwe. A practical architecture for an anycast cdn. *ACM Trans. Web*, 5(4):17:1–17:29, Oct 2011.

[3] H. Ballani, P. Francis, and S. Ratnasamy. A Measurement-based Deployment Proposal for IP Anycast. In *Proc. ACM IMC*, 2006.

[4] B. Barber, M. Larson, and M. Kosters. Traffic Source Analysis of the J Root Anycast instances. Nanog, 2006.

[5] I. Bermudez, M. Mellia, M. Munafò, R. Keralapura, and A. Nucci. DNS to the Rescue: Discerning Content and Services in a Tangled Web. ACM IMC, 2012.

[6] P. Boothe and R. Bush. DNS Anycast Stability: Some Early Results. CAIDA, 2005.

[7] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the Performance of an Anycast CDN. In *Proc. ACM IMC*, 2015.

[8] D. Cicalese, J. Augé, D. Joumblatt, T. Friedman, and D. Rossi. Characterizing IPv4 Anycast Adoption and Deployment. In *Proc. CoNEXT*, 2015.

[9] D. Cicalese, D. Giordano, A. Finamore, M. Mellia, M. Munafò, D. Rossi, and D. Joumblatt. A First Look at Anycast CDN Traffic. *ArXiv e-prints*, 2016.

[10] D. Cicalese, D. Joumblatt, D. Rossi, M.-O. Buob, J. Augé, and T. Friedman. A Fistful of Pings: Accurate and Lightweight Anycast Enumeration and Geolocation. In *Proc. IEEE INFOCOM*, 2015.

[11] L. Colitti. Measuring Anycast Server Performance: The Case of K-root. Nanog, 2006.

[12] X. Fan, J. S. Heidemann, and R. Govindan. Evaluating Anycast in the Domain Name System. In *Proc. IEEE INFOCOM*, 2013.

[13] A. Finamore, M. Mellia, M. Meo, M. Munafò, and D. Rossi. Experiences of Internet Traffic Monitoring with Tstat. *Network, IEEE*, 25:8–14, 2011.

[14] A. Flavel, P. Mani, D. A. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev. FastRoute: A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs. In *Proc. USENIX NSDI*, 2015.

[15] T. Hardie. Known Content Network (CN) Request-Routing Mechanisms. IETF RFC 3568, 2003.

[16] D. Karrenberg. Anycast and BGP Stability: A Closer Look at DNSMON Data. Nanog, 2005.

[17] D. Katabi and J. Wroclawski. A Framework for Scalable Global IP-anycast (GIA). In *Proc. ACM SIGCOMM*, 2000.

[18] M. Levine, B. Lyon, and T. Underwood. Operational Experience with TCP and Anycast. Nanog, 2006.

[19] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. C. Claffy. Two Days in the Life of the DNS Anycast Root Servers. In *Proc. of PAM*, 2007.

[20] D. Madory, C. Cook, and K. Miao. Who Are the Anycasters. Nanog, 2013.

[21] S. Sarat, V. Pappas, and A. Terzis. On the Use of Anycast in DNS. In *Proc. IEEE ICCCN*, 2006.

[22] M. Trevisan, F. Alessandro, M. Mellia, M. Munafò, and D. Rossi. DPDKStat: 40Gbps Statistical Traffic Analysis with Off-the-Shelf Hardware. *In Tech. Rep., 2016*.